# Dr. Who's Dialog System Overview

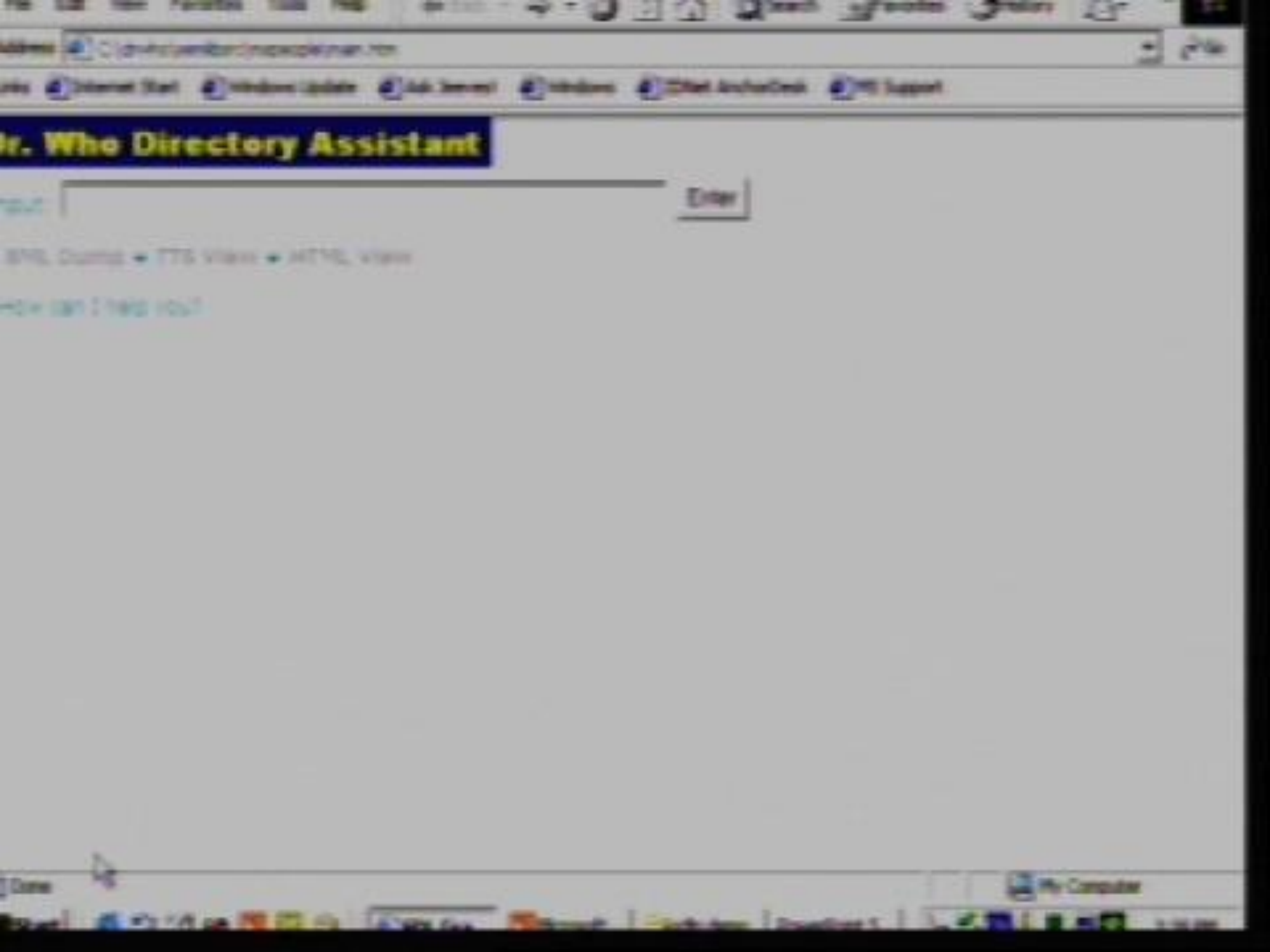
Kuansan Wang
Speech Technology Group
Microsoft Research

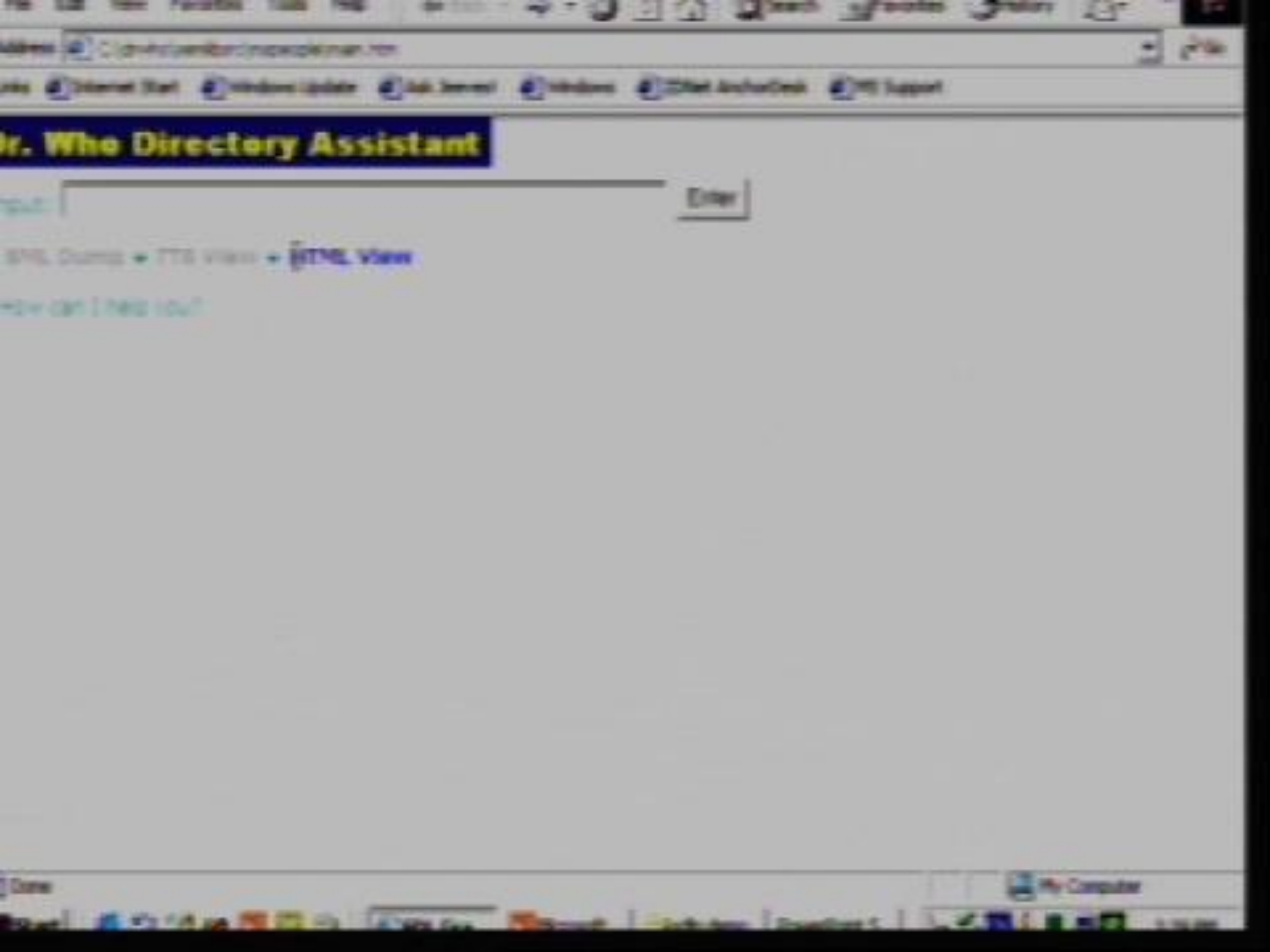
# Outline

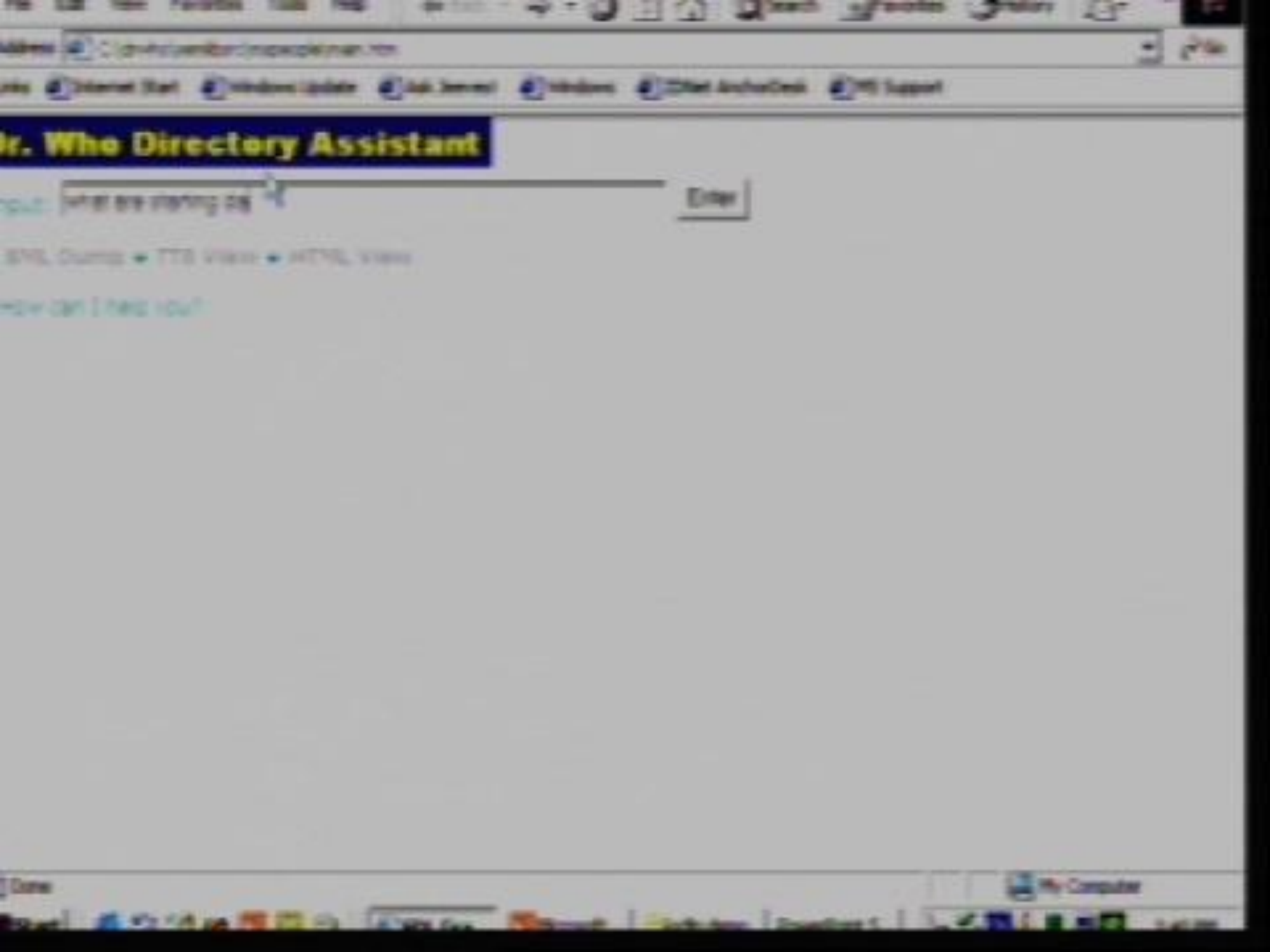
- Goals of STG's dialog efforts
- Demo
  - Dr. Who Directory Assistant
- Problem definition and solution
- Architecture
- Implementation
- What W3C is doing

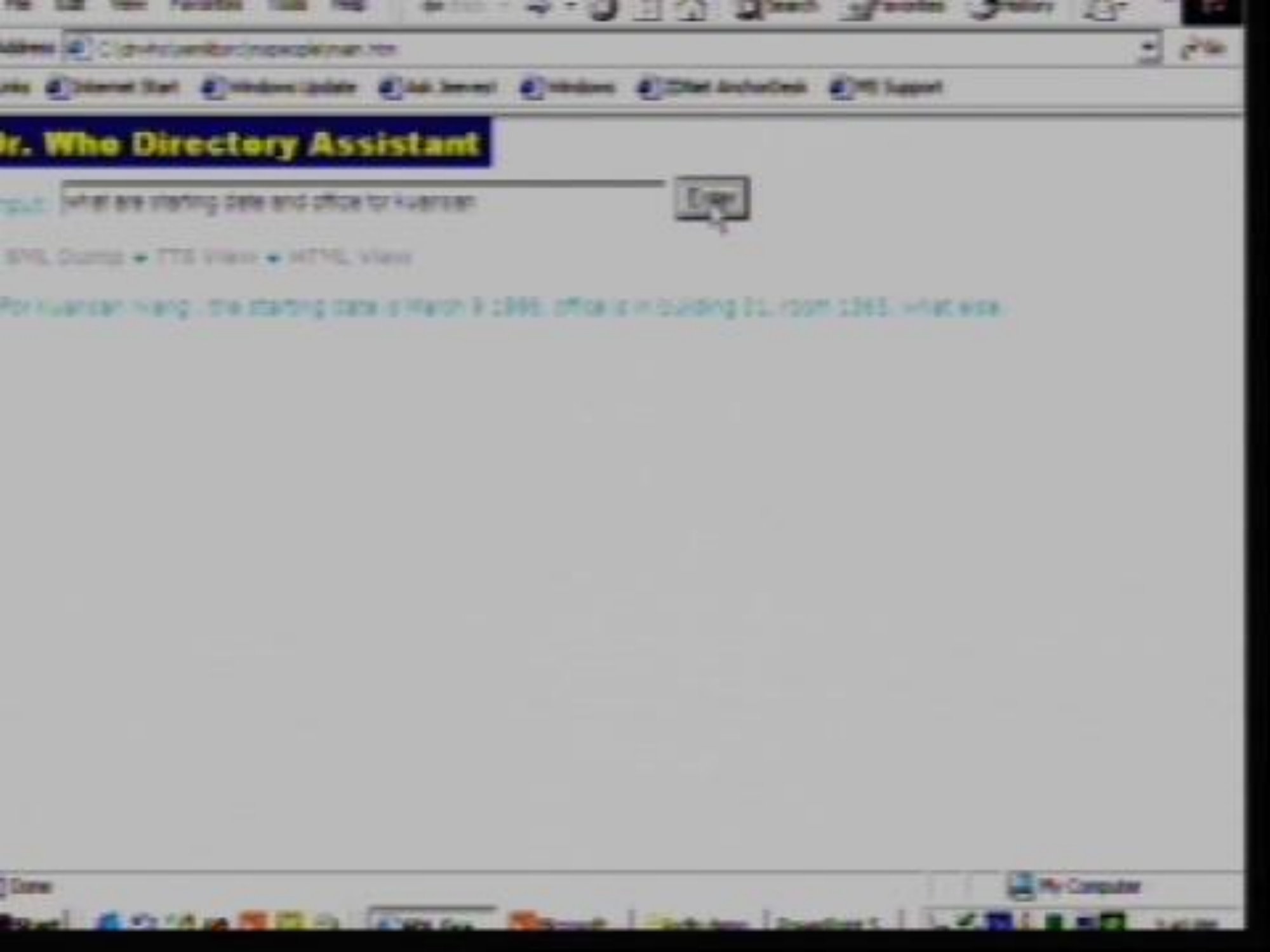
# Design Goals

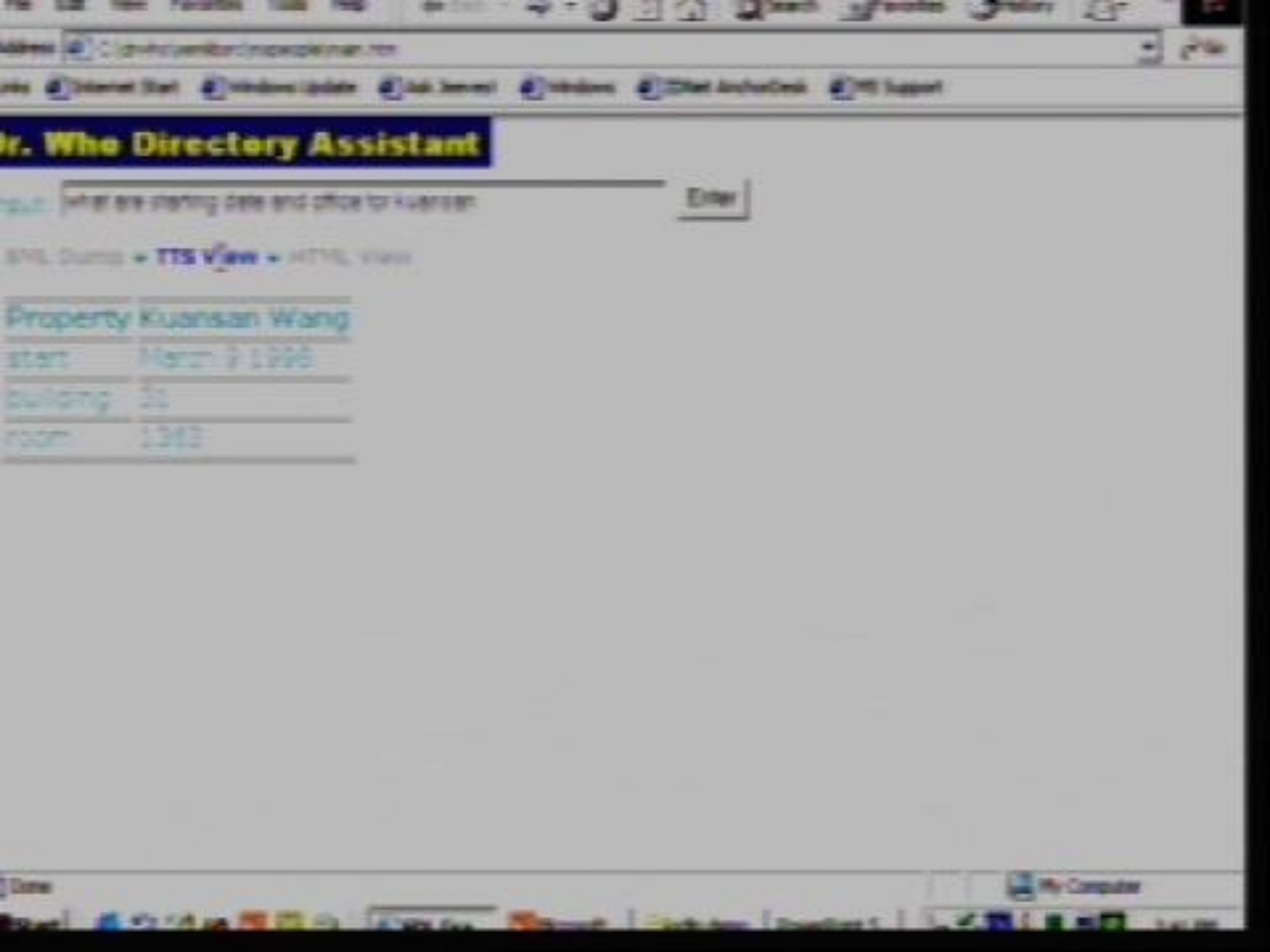
- Universal Access
  - Multimodal
  - Seamless integration and switching
- Intelligent behavior
  - Reasoning like an expert
  - Adaptive dialog strategies
  - Dynamic semantic/language models
- Simple architecture
  - Easy to implement, max reusability

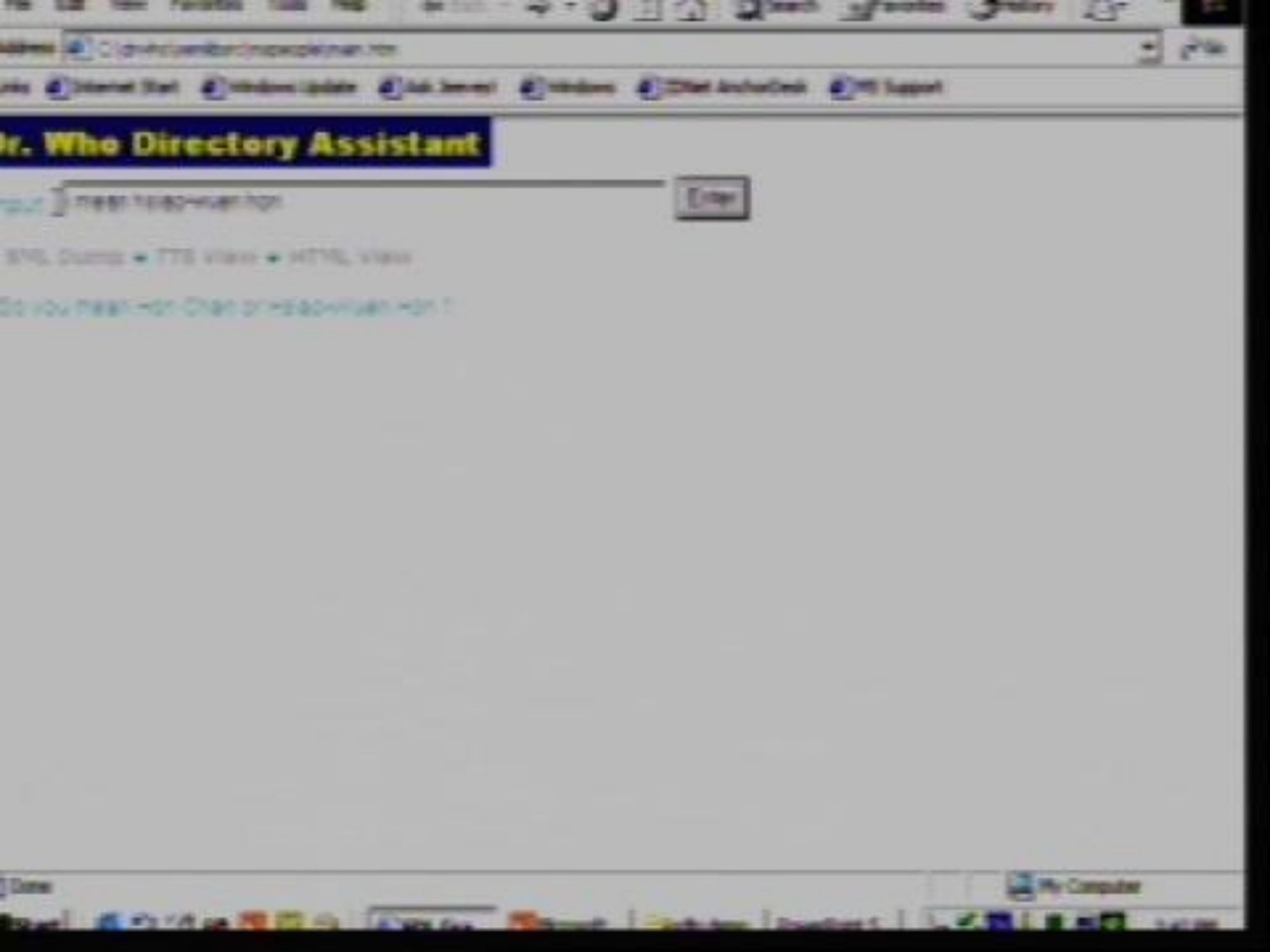

r. Who Directory Assistant
Enter

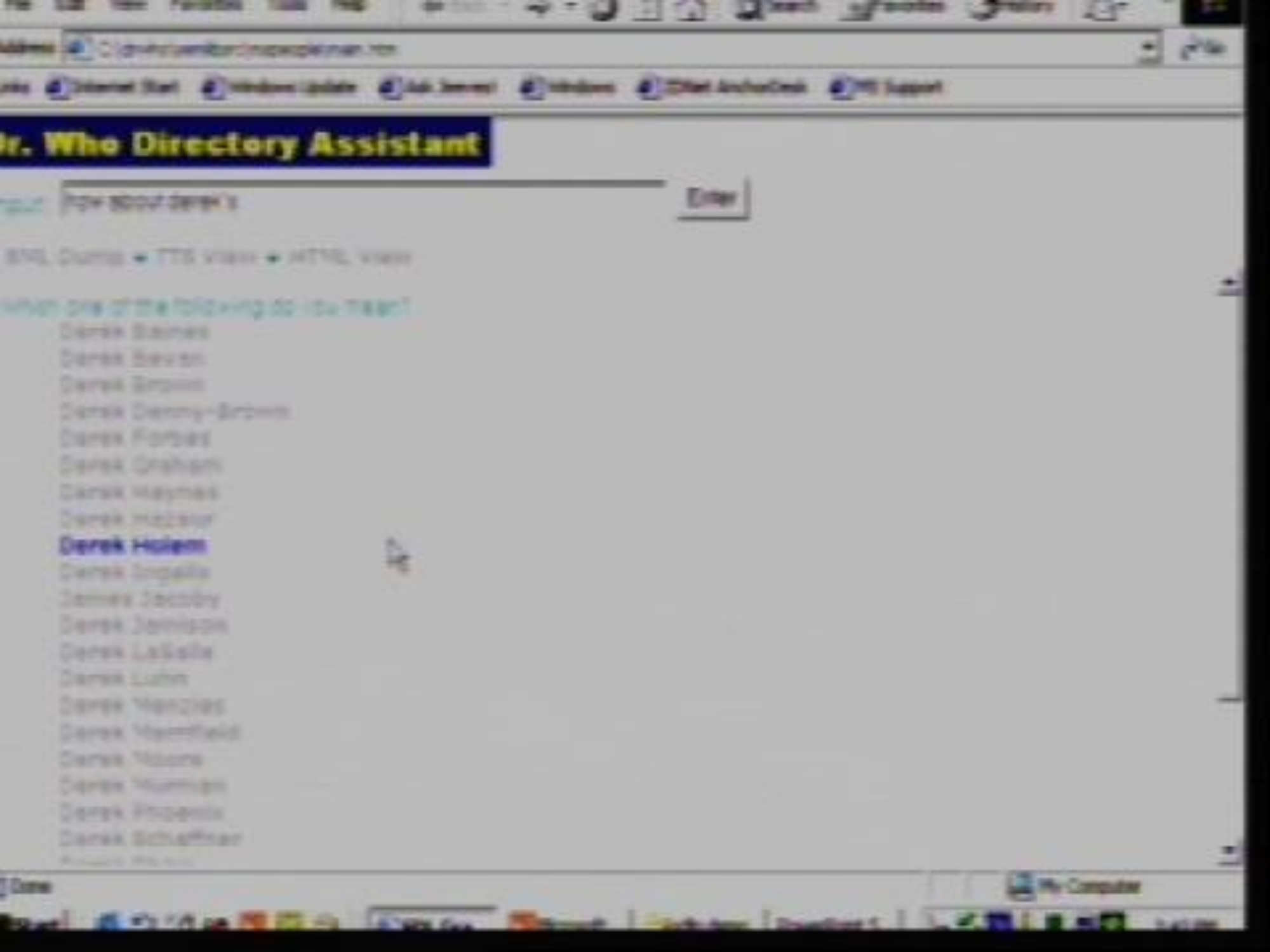

Dr. Who Directory Assistant
Enter
HTML View


Dr. Who Directory Assistant
Enter

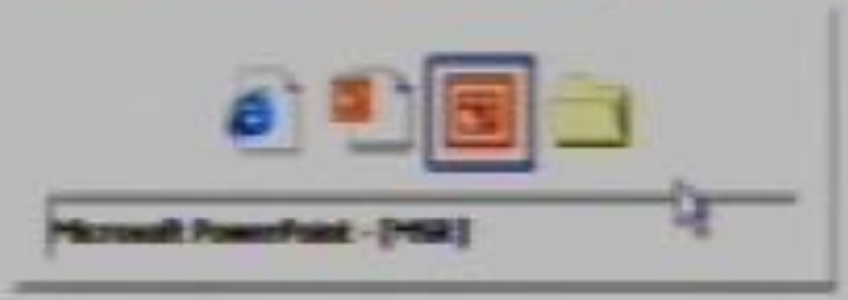

Dr. Who Directory Assistant


Dr. Who Directory Assistant
Enter
Property Kuansan Wang


Dr. Who Directory Assistant


r. Who Directory Assistant
Enter
Derek Holem

Dr. Who Directory Assistant



# Design Goals

- Universal Access
  - Multimodal
  - Seamless integration and switching
- Intelligent behavior
  - Reasoning like an expert
  - Adaptive dialog strategies
  - Dynamic semantic/language models
- Simple architecture
  - Easy to implement, max reusability

# What is semantics?

# What is semantics?

- Descriptive semantics
  - Detailed description of objects, actions, events

# What is semantics?

- Descriptive semantics
  - Detailed description of objects, actions, events
- Equivalence semantics
  - Predicate calculus
- Procedural semantics
  - Meaning defined by actions

# What is semantics?

- Descriptive semantics
  - Detailed description of objects, actions, events
- Equivalence semantics
  - Predicate calculus
- Procedural semantics
  - Meaning defined by actions
- STG's approach: all of above

# Dialog: A Pattern Recognition Problem!

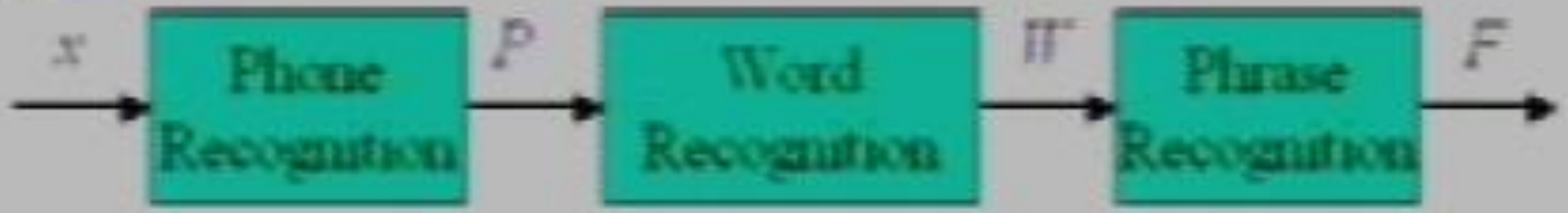
Objective $A_{opt} = \arg\max_A P(A|x, S_{n-1})$

$= \arg\max_A \Sigma_S P(A|S_n) \Sigma_F P(S_n|F, S_{n-1}) P(F|x)$

(Viterbi) $= \arg\max_{A,S} P(A|S_n) \Sigma_F P(S_n|F, S_{n-1}) P(F|x)$

- $A$ proper actions
- $S_n$ discourse semantics for the n-th turn
- $F$ abstract representation of signal $x$

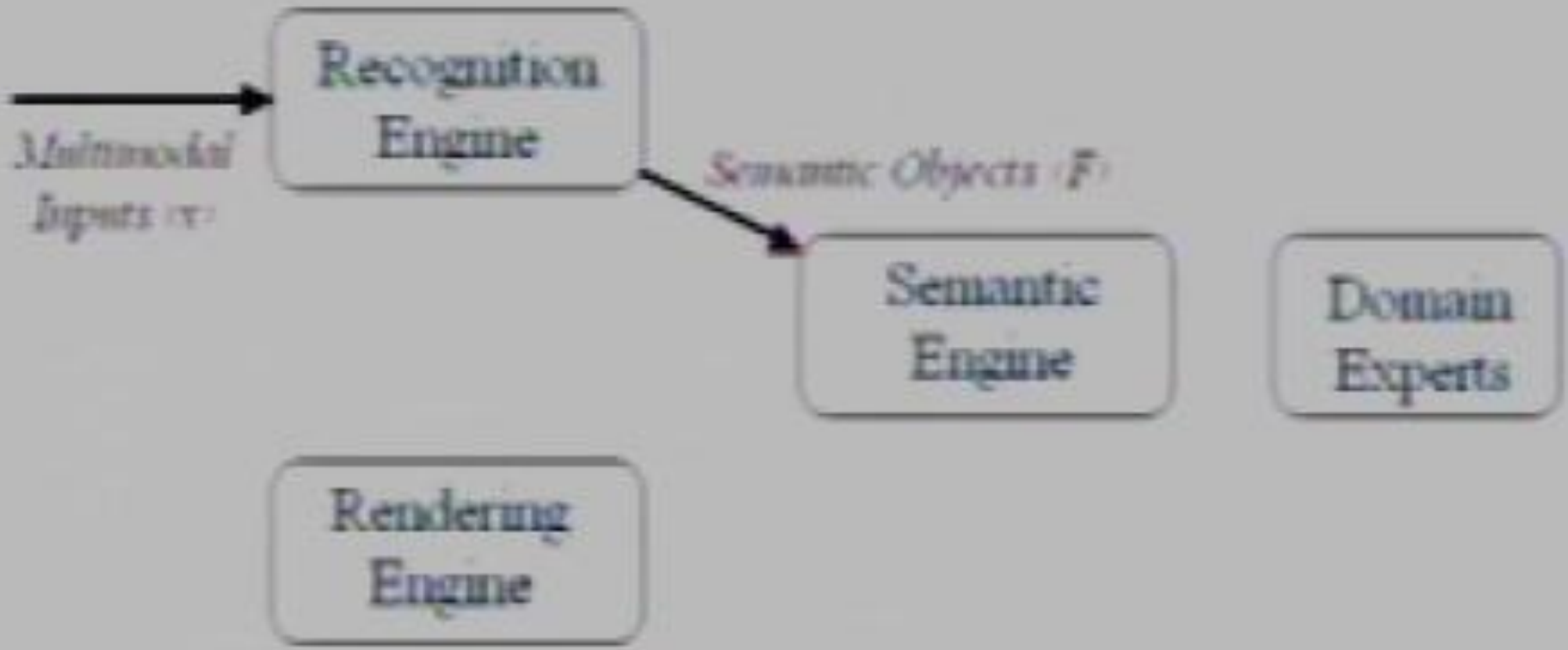
# Analogy to Speech Recognition



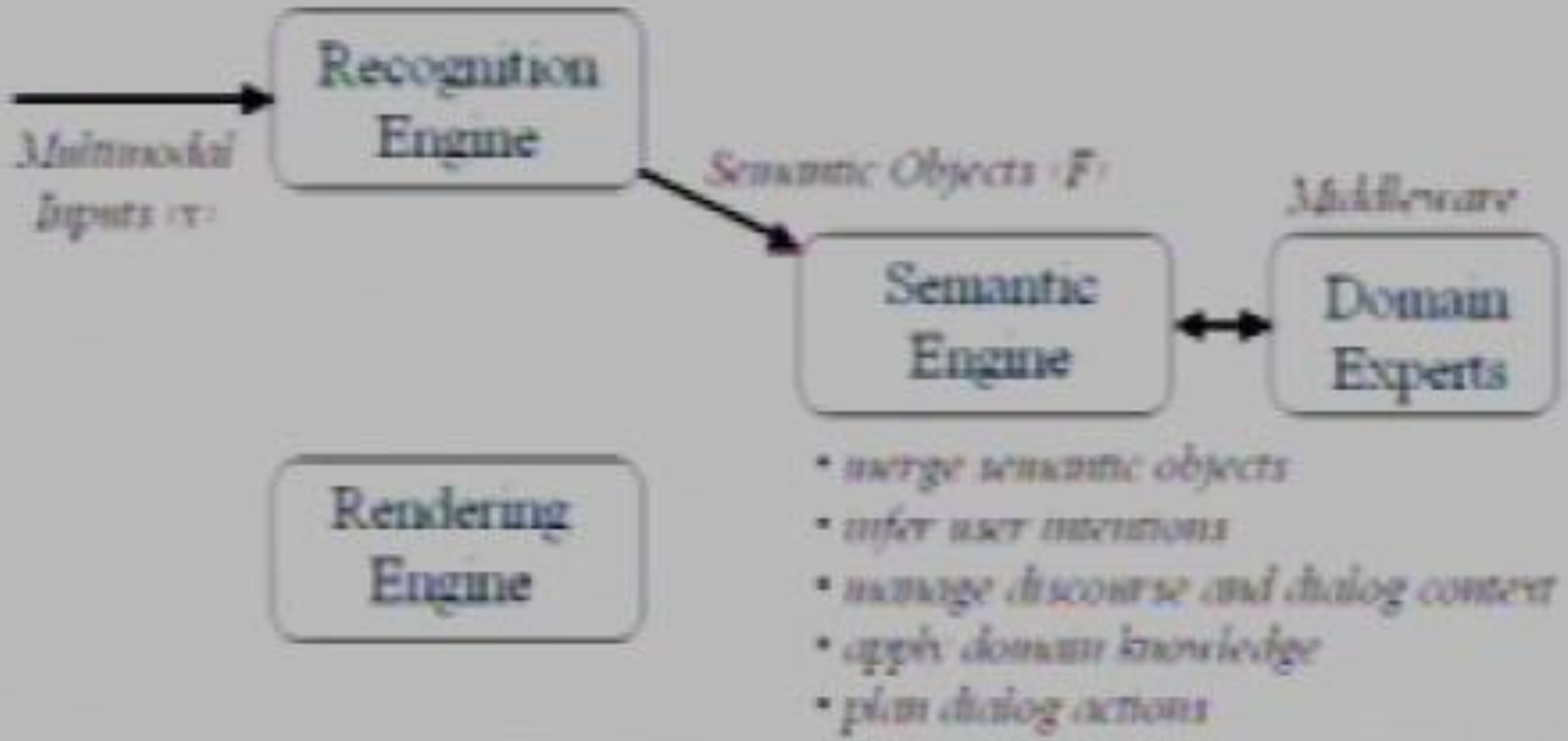

Key components for each block:

- Engine: implement dynamic search for arg max
- Model: application specific probabilistic measures

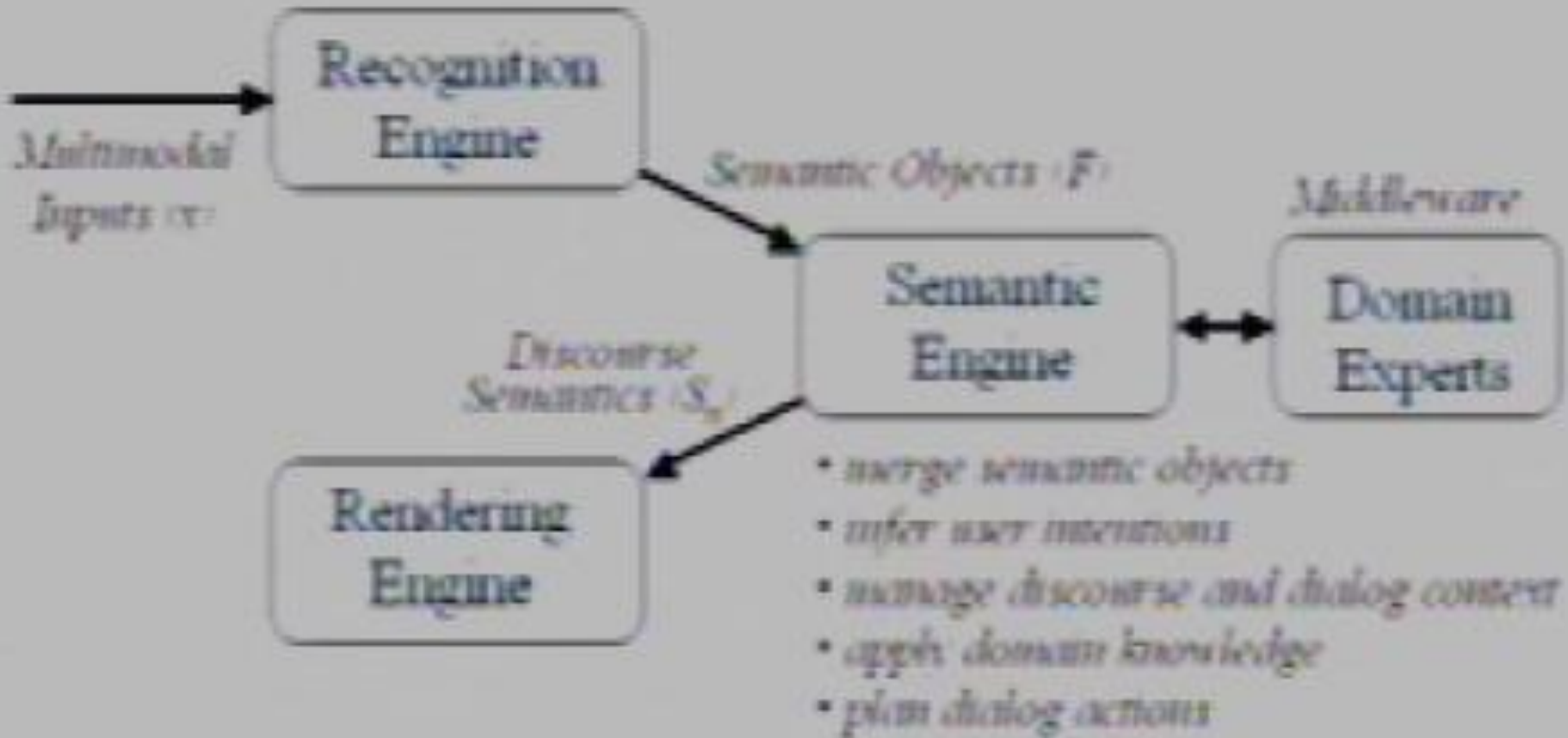
# Architecture



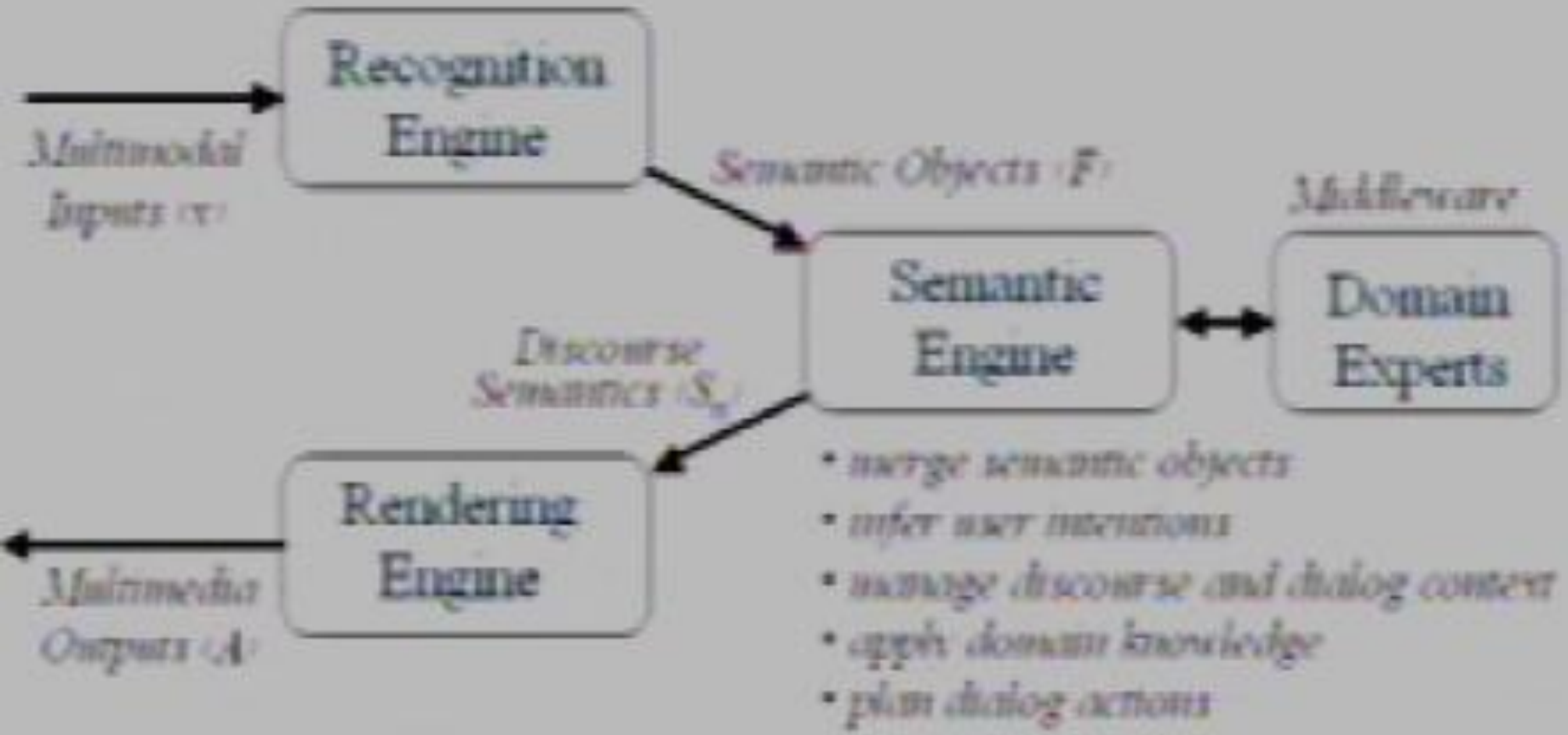
# Architecture



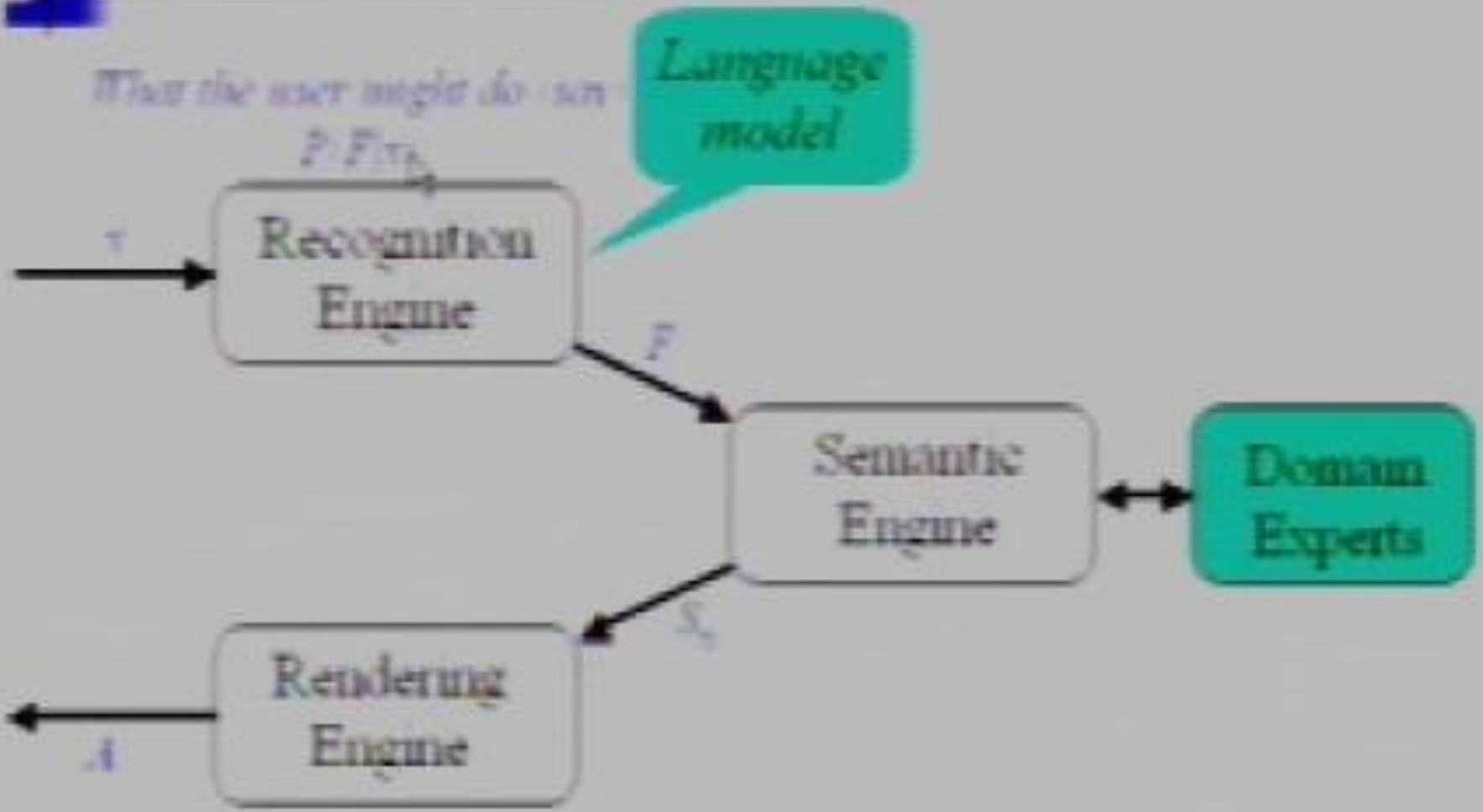
# Architecture



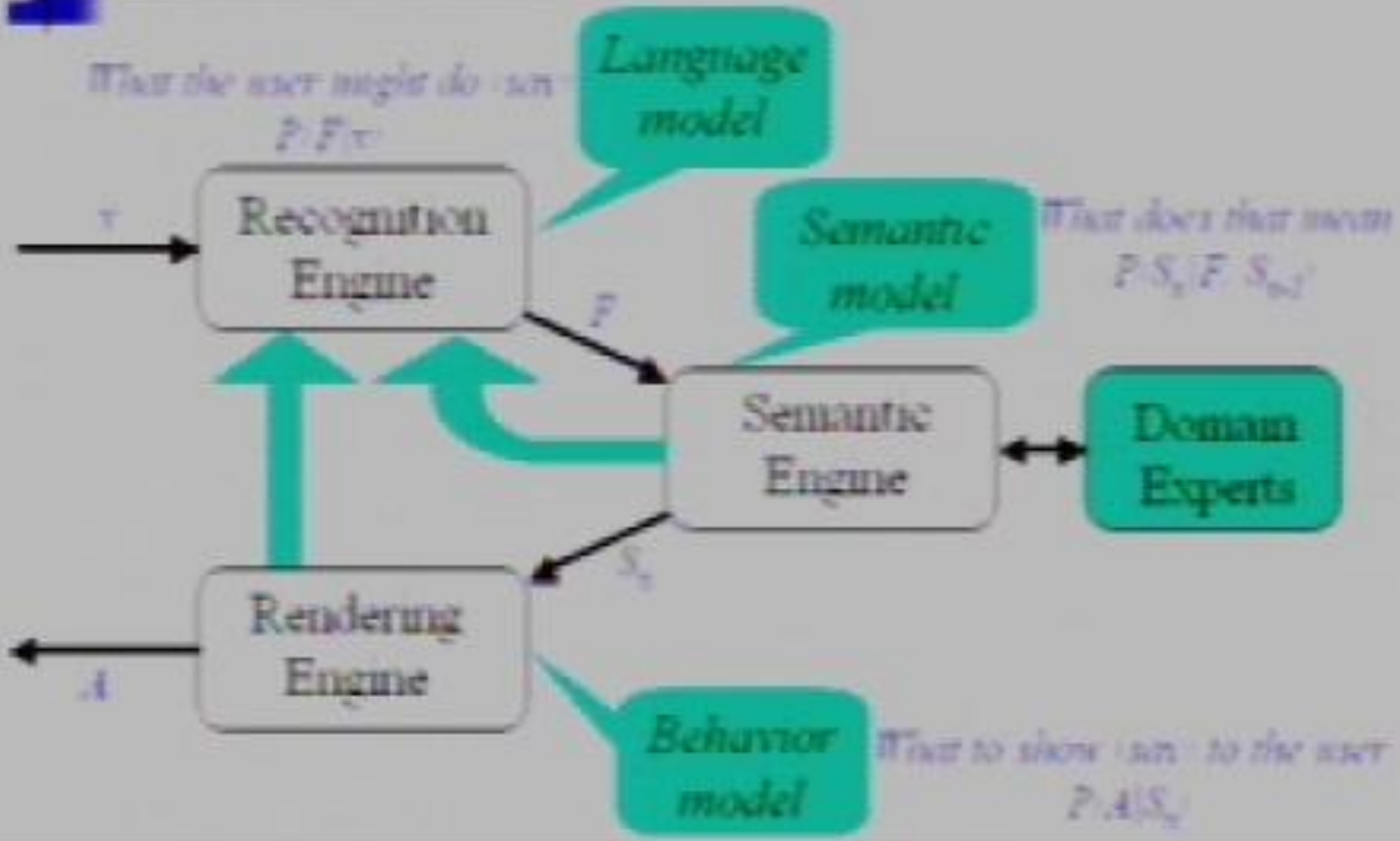
# Architecture



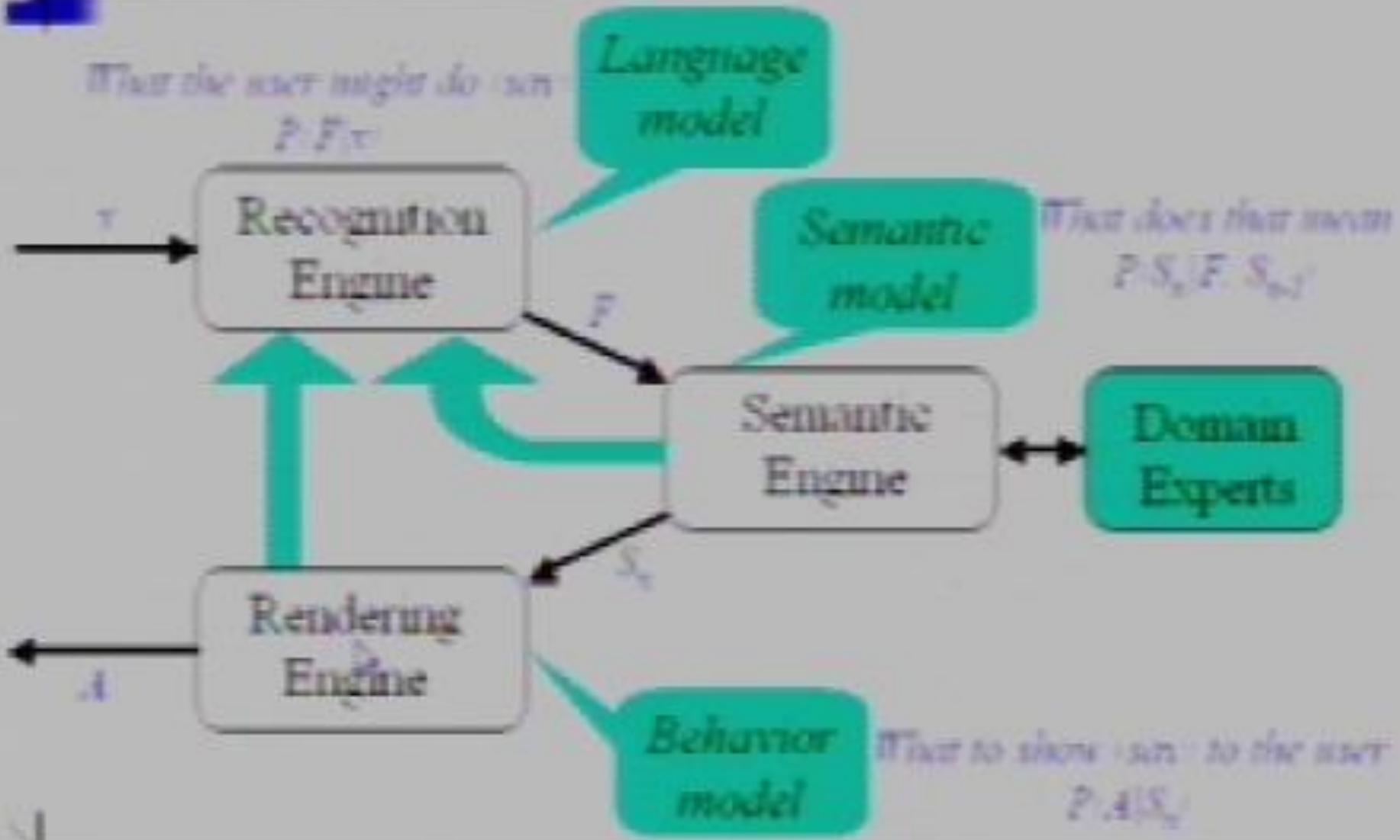
# Application Grammars



# Application Grammars



# Application Grammars



# Knowledge Representation

- Assume app has well-defined schema, e.g., in Entity Relation Diagram (ERD)
  - Physical entities
    - object/data models
  - Functional entities
    - event/action models
- App schema may be intended mainly for machine-machine communications
- Command & Control: speech interface to app schema

# NL Semantics

- NL is powerful and easier to use
  - "send it to those in the design meeting last week"
- NL can be imprecise
  - Which message?
    - Anaphora resolution (from context)
  - Who are the recipients?
    - Deixis resolution
  - What meeting?
    - Ellipsis resolution

# NL Semantics

- NL is powerful and easier to use
  - "send it to those in the design meeting last week"
- NL can be imprecise
  - Which message?
    - Anaphora resolution (from context)
  - Who are the recipients?
    - Deixis resolution
  - What meeting?
    - Ellipsis resolution

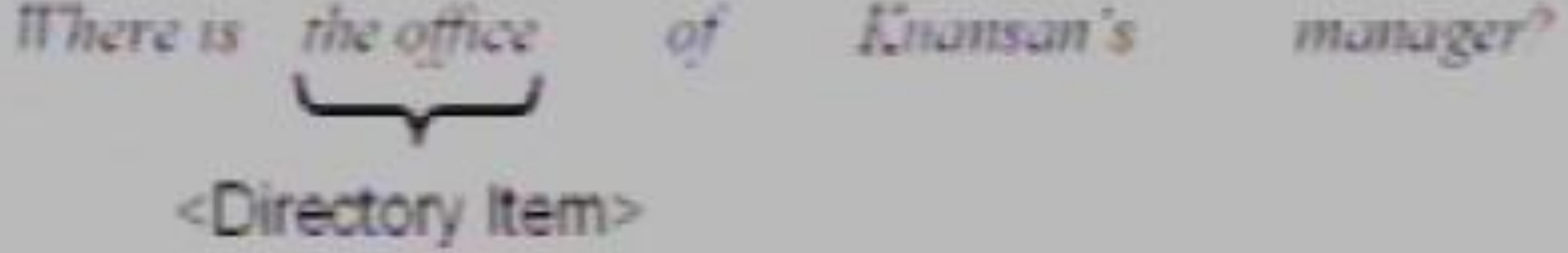
# Semantic Objects

- Abstraction of domain
  - Equivalence semantics
  - Shield NL developers from schema changes
- Encapsulation:
  - How to identify
    - manners of realization (e.g. CFG for speech/text)
  - How to understand
    - links to app schema (e.g. Entity type)
  - How to deal with exceptions
    - dialog actions
- Inheritance
- Polymorphism

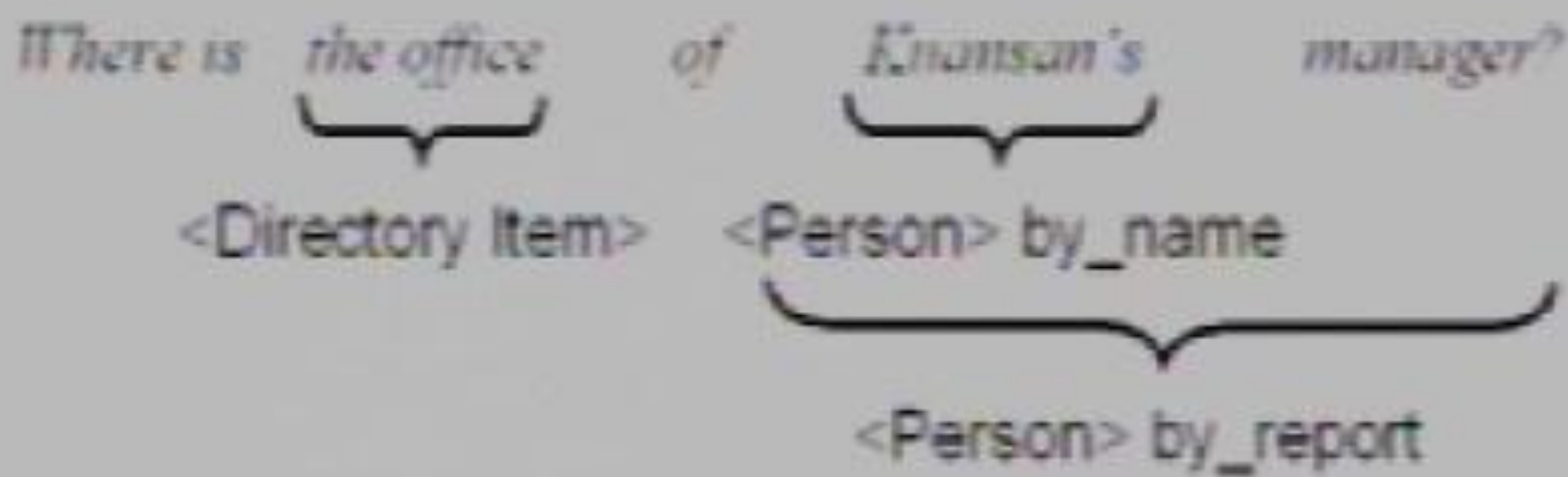
# Polymorphism

- Variations in schemas
  - Attributes vary for the same entity type
  - NGWS: schema recognition and conversion services
- Many ways to indicate the same thing
  - By full name: *Xuedong Huang*
  - By relationship: *Kuansan's manager*
  - By nickname: *X. D.*
  - By clicking XD's picture on the screen
- Each semantic object represents one way to identify an entity
- Semantic evaluation uses entity types
  - Ex. Send mail to <person>

# Parse: semantic object tree



# Parse: semantic object tree



# Semantic Engine

- Semantic object forest ($S_{n-1}$):
  - Attach parse tree ($F$) to current discourse tree
  - Resolve semantic objects into entities
- Entity Memory
  - Associative chunking memory: typed priority queue
- Reference Resolution
  - Based on entity memory
  - Resolve anaphora, ellipsis, and deixis in the same manner

# Models & Concepts Annotation – Use of Markup Languages

- Language model
  - SAPI format
- Semantic objects (output of recognition engine)
  - SML, semantic markup language, derived from XML

# Semantic object as an SML instance

SML Instance

itinerary

origin

city Seattle city

state Washington state

origin

destination

city Portland city

state Maine state

destination

date 07 04 2000 date

itinerary

# Semantic object as an SML instance



```
SML Instance
 itinerary
  origin
   city Seattle city
   state Washington state
  origin
  destination
   city Portland city
   state Maine state
  destination
  date 07 04 2000 date
 itinerary
```

- ML Schema:
  - What are legitimate tags
  - What does a tag contain?

# XForm

- W3C HTML WG
- Schema definition language for HTML forms
- Separation of contents from presentation
- Status: early draft
  - Data model
- W3C Voice Browser WG:
  - Form filling metaphor
  - Candidate for semantic representation

# SML schema in XForm

SML Instance

```
itinerary
  origin
    city Seattle city
    state Washington state
  origin
  destination
    city Portland city
    state Maine state
  destination
  date 07 04 2000 date
itinerary
```

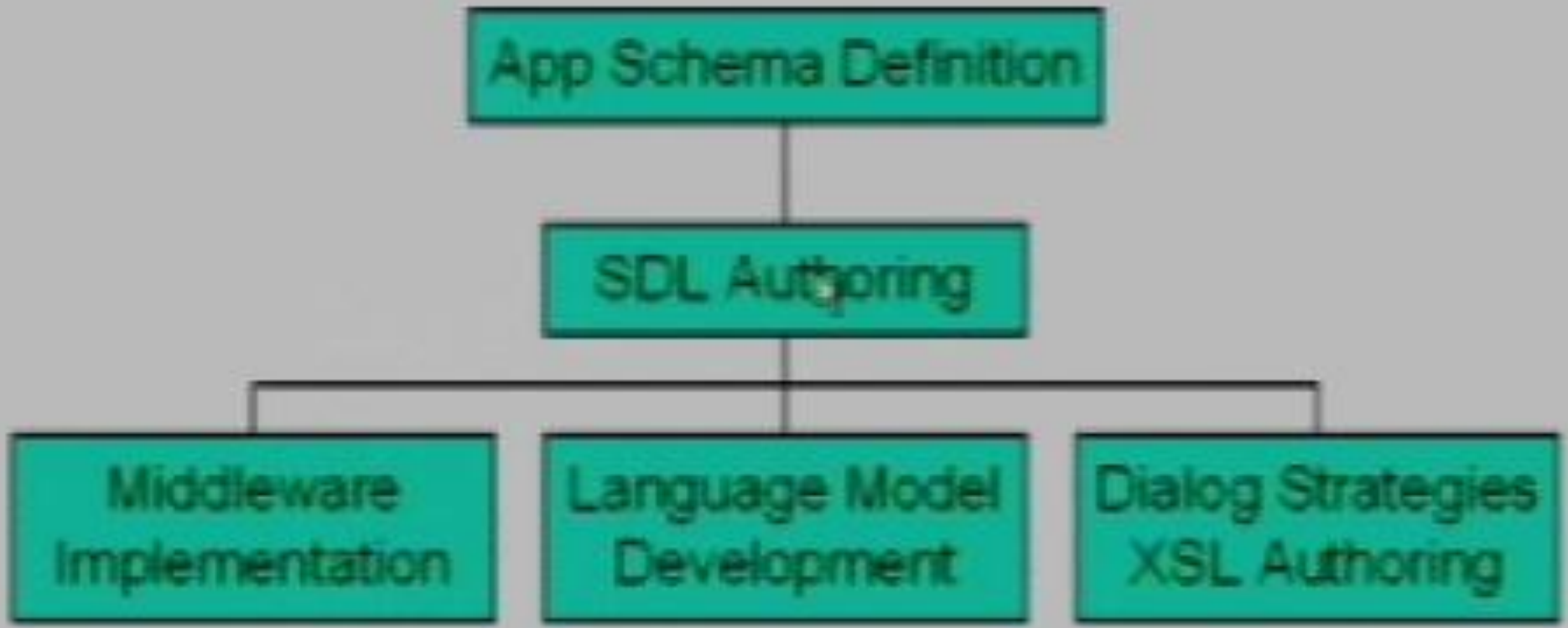
W3C XForm Schema

```
group name="itinerary"
  group name="origin"
    string name="city"
    string name="state"
  group
  group name="destination"
    string name="city"
    string name="state"
  group
  date name="date"
group
```

# Semantic Definition Language

- STG's SML schema definition language
  - Defines constructs of semantic objects (semantic classes) for recognition engine
  - Defines relationships between semantic objects and entities for semantic engine
  - Defines discourse semantics for rendering engine
- Tailored to NL and multimodal applications
  - Supports semantic polymorphism
  - Supports hierarchical nesting of unknown levels in depth and recursion
  - Declarative cues for reference resolution

# App Development Process



# Response Rendering in XSLT

- XSLT: already W3C standard
- Basic usage:

```
<xsl:template match="pattern in source doc">
  ... outputs in targeted ML
  <xsl:apply-templates select="pattern" />
</xsl:template>
```

- XSLT: transforming SML to targeted ML (e.g. HTML, SAPI/TTS ML)
- Prolog-like programming paradigm suitable for implementing intelligent dialog strategies

# Summary

- Dialog is a pattern recognition problem
- Three sub-problems
  - Signal recognition
  - Discourse semantics
  - Action rendering
- Notion of semantic objects for NL
  - Combine descriptive, procedural semantics through equivalence semantics
- Separating contents from presentation
  - Seamless multimodal universal access
  - Adaptive behavior
  - Simple architecture and engineering benefits